

An algebra for signal processing

Henning Thielemann

Institut für Informatik, Martin-Luther-Universität Halle-Wittenberg, Germany
 henning.thielemann@informatik.uni-halle.de

Abstract. Our paper presents an attempt to axiomatise signal processing. Our long-term goal is to formulate signal processing algorithms for an ideal world of exact computation and prove properties about them, then interpret these ideal formulations and apply them without change to real world discrete data. We give models of the axioms that are based on GAUSSIAN functions, that allow for exact computations and automated tests of signal algorithm properties.

Keywords: Algebra, Signal processing, Gaussian function

1 Introduction

1.1 Motivation

In signal processing we consider real or complex valued functions. These functions represent signals or frequency spectra and their arguments are considered to be time values or frequency values, respectively. There are some fundamental operations like pointwise multiplication “.” and convolution “*” of signals. The FOURIER transform \mathcal{F} converts between signals and frequency spectra. For precise definitions of these operations see Section 2. Additionally there are some essential laws, that every signal processing scientist is familiar with, such as the law, that the FOURIER transform is an homomorphism, that maps multiplication to convolution:

$$\mathcal{F}(x \cdot y) = \mathcal{F}x * \mathcal{F}y.$$

This is an important connection and it is amazingly simple, but it is not quite true.

- The first problem is, that depending on the precise definitions of the involved operations there may be a factor 2π or $\sqrt{2\pi}$ to make the above identity correct. This is like working in the imperial system, where a gallon is not just the cubic power of a length unit but 231 cubic inch. This is at least cumbersome and error-prone when done manually, but it also complicates computer implementations. With the transcendent factors we have to work in fields like $\mathbb{Q}(\sqrt{2\pi})$ or $\mathbb{Q}(\pi)$ or would have to work with approximations, but working with only rational numbers would of course be still exact and more efficient. However, if we succeed to suppress the transcendental factor in the above identity by the change of a definition, then it will show up in another place.

Thus our first contribution is to select a set of operations that are common in signal processing and list identities that show their interrelation in principle. We use the degrees of freedom in the operation definitions for simplifying the laws as much as possible, that is, avoid constant factors and assert maximum symmetry in Section 2. Metaphorically speaking, we try to define something like a metric system for signal processing.

- The second problem is, that the convolution and FOURIER transform are not always defined. For instance the straightforward definitions of the FOURIER transform and its inverse as given in Section 2, map from absolutely integrable functions to bounded functions, i.e. $\mathcal{F}, \mathcal{F}^{-1} : \mathcal{L}_1(\mathbb{R}) \rightarrow \mathcal{L}_\infty(\mathbb{R})$. That is, in general you cannot apply the inverse FOURIER transform to the result of a FOURIER transform. There are even more such difficulties, but they can be resolved by restriction to the SCHWARTZ space \mathcal{S} of arbitrarily smooth and rapidly decaying functions.
- The third problem is, that commonly signal processing in a computer is performed on discretised functions with finite precision numbers. That is, laws as the above one do not hold exactly.

Our contribution concerning this problem is in the subsections of Section 3. There we take the presented laws as axioms and develop models in terms of extended GAUSSIAN functions. All of these functions are in \mathcal{S} . The more we extend those functions by factors and parameters, the more operations we can support. Our goal is to give definitions of functions, such that we only need to cope with rational parameters. This way we can represent a wide range of functions, we can compute exactly and we could even generalise these models to any algebraic field.

1.2 Basics

The theory of Algebra provides many axiomatically described structures, that generalise common mathematical objects and operations on them. E.g. groups abstract permutations, lattices abstract BOOLEAN logic, rings abstract integer arithmetic, fields abstract rational arithmetic. If we are able to perform a proof with the axioms of a particular algebraic structure, then our proof automatically applies to every such structure. In this paper we would like to abstract from what is commonly called signal processing.

Signal processing is the research area of construction, transformation and analysis of oscillation functions in one variable (the time), approximation of those functions and related operations with discretised data and efficient implementations on digital computers. An important view on an audio signal is the frequency spectrum, because that is closer to the way humans hear than the time-domain representation of a signal. We obtain a frequency spectrum by the FOURIER transform of a signal. Closely related to signal processing is image processing, with the main difference being, that functions in two variables are studied. Another related area are random distributions in stochastics. The convolution, a common operation in signal processing, of two random distributions yields the distribution of the sum of two random variables. The FOURIER

transform yields the characteristic function of a random distribution and there are more connections (see Section 3).

However, signal processing, image processing and random distributions are relatively seldom viewed from an algebraic point of view. This may have to do with the tradition to focus on signal values (e.g. $f(t)$) rather than on larger objects like signals (e.g. f). E.g. it is common to express the delay of a signal by $f(t - d)$ rather than to use an operator like in $f \rightarrow d$. On the one hand this has the advantage, that some properties are obvious (e.g. $f(t - (d_0 + d_1)) = f((t - d_0) - d_1)$), since they are a consequence of simple arithmetic, but are not obvious at the higher level (i.e. $f \rightarrow (d_0 + d_1) = (f \rightarrow d_1) \rightarrow d_0$). On the other hand we cannot cleanly express identities involving intrinsically functional transforms like the FOURIER transform, that has lead to custom notations like $F(\omega) \xleftrightarrow{\mathcal{F}} f(t)$, that could be expressed the functional way as $\mathcal{F}F = f$.

2 Finding a set of fundamental operations

We like to start this section listing the operations, that are commonly used in signal processing, together with typical applications. We begin with the operations with indisputable definitions and continue with the ones, where differing definitions are around in the literature. For the variant definitions we show, what laws they imply. Then we choose the definitions that make the laws most simple. We close the section with a comprehensive list of laws that hold for our definitions.

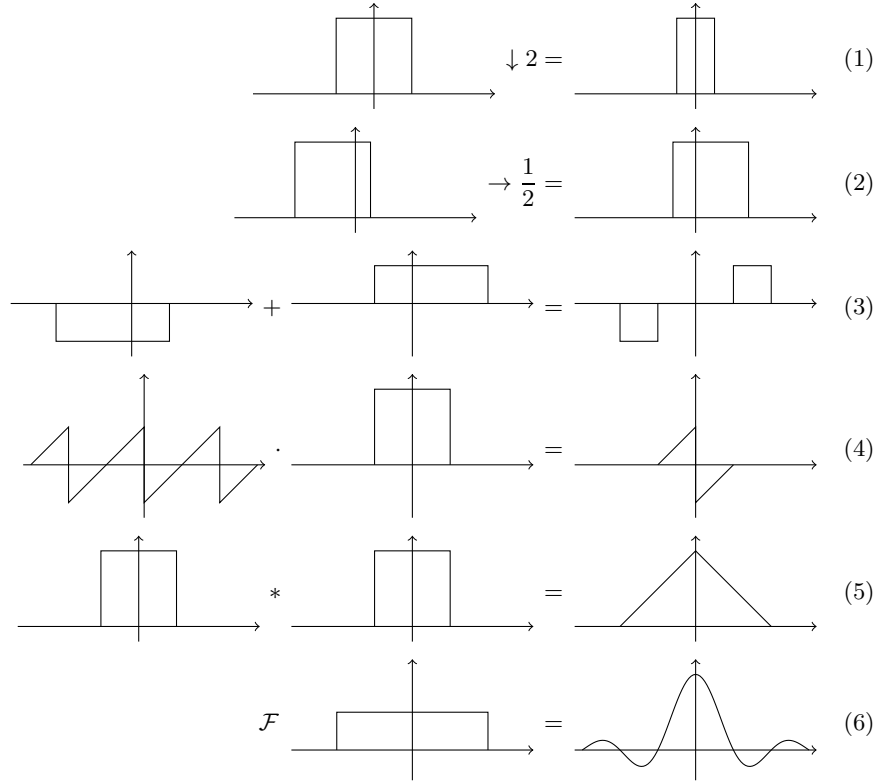
In Table 1 we list the signal operations together with their definitions and in Figure 1 we illustrate them using example signals.

- Shrinking a signal as in (1) means to increase all contained frequencies proportionally and to shorten time accordingly.
- Translating a signal as in (2) means to delay it.
- Summing two signals as in (3) means to superpose them, that is, to play them simultaneously.
- Multiplying two signals as in (4) means to control the amplitude of one signal by the other one. For certain choices of signals this is also known as ring-modulation.
- Convolution two signals as in (5) means to apply the sound characteristics of one signal to the other one. It may be used for suppressing or emphasising certain frequencies, for smoothing or for application of reverb.

We continue with the controversial definitions. Actually, there is essentially one operation, that is defined differently throughout the signal processing literature: The FOURIER transform. It is the transform that computes the frequency spectrum for a signal in the time domain (*backward* or *analysis* transform) and vice versa (*forward* or *synthesis* transform). For an example see (6) in Figure 1. The other definition with varying instances is the one for functions that are eigenfunctions of the FOURIER transform. Obviously it depends entirely on the definition of the FOURIER transform.

Table 1. *Basic signal processing operations*

operation	application	definition
shrinking	alter pitch and time	$(x \downarrow k)(t) = x(k \cdot t)$
translate	delay	$(x \rightarrow k)(t) = x(t - k)$
adjoint		$x^* = \overline{x} \downarrow -1$
sum	mixing	$(x + y)(t) = x(t) + y(t)$
multiplication	envelope	$(x \cdot y)(t) = x(t) \cdot y(t)$
convolution	frequency filter	$(x * y)(t) = \int_{\mathbb{R}} x(\tau) \cdot y(t - \tau) \, d\tau$
modulation	frequency shift	$x \cdot \text{cis1} \quad \text{where } \text{cis1 } t = \exp(2\pi i \cdot t)$

**Fig. 1.** *Illustration of the basic signal processing operations.*

1. Oscillations with period 1

$$\begin{array}{ll}
\text{FOURIER forward} & \mathcal{F}_1 x(\tau) = \int_{\mathbb{R}} \exp(2\pi i \cdot \tau \cdot t) \cdot x(t) \, dt \\
\text{FOURIER backward} & \mathcal{F}_1^{-1} x(\tau) = \int_{\mathbb{R}} \exp(-2\pi i \cdot \tau \cdot t) \cdot x(t) \, dt \\
\text{duality} & \mathcal{F}_1(\mathcal{F}_1 x) = x \downarrow -1 \\
\text{eigenfunction} & g(t) = \exp(-\pi \cdot t^2) \\
\text{eigenvalue is 1} & \mathcal{F}_1 g = g \\
\text{unitarity} & \langle x, y \rangle = \langle \mathcal{F}_1 x, \mathcal{F}_1 y \rangle \\
\text{convolution theorem} & \mathcal{F}_1(x * y) = \mathcal{F}_1 x \cdot \mathcal{F}_1 y \\
& \mathcal{F}_1(x \cdot y) = \mathcal{F}_1 x * \mathcal{F}_1 y \\
\text{derivative} & \mathcal{F}_1(x') = -2\pi i \cdot \text{id} \cdot \mathcal{F}_1 x
\end{array} \tag{7}$$

2. Oscillations with period 2π

$$\begin{array}{ll}
\text{FOURIER forward} & \mathcal{F}_2 x(\tau) = \frac{1}{\sqrt{2\pi}} \cdot \int_{\mathbb{R}} \exp(i \cdot \tau \cdot t) \cdot x(t) \, dt \\
\text{FOURIER backward} & \mathcal{F}_2^{-1} x(\tau) = \frac{1}{\sqrt{2\pi}} \cdot \int_{\mathbb{R}} \exp(-i \cdot \tau \cdot t) \cdot x(t) \, dt \\
\text{duality} & \mathcal{F}_2(\mathcal{F}_2 x) = x \downarrow -1 \\
\text{eigenfunction} & g(t) = \exp(-t^2)
\end{array} \tag{9}$$

$$\begin{array}{ll}
\text{eigenvalue is 1} & \mathcal{F}_2 g = g \\
\text{unitarity} & \langle x, y \rangle = \langle \mathcal{F}_2 x, \mathcal{F}_2 y \rangle \\
\text{convolution theorem} & \mathcal{F}_2(x * y) = \sqrt{2\pi} \cdot \mathcal{F}_2 x \cdot \mathcal{F}_2 y
\end{array} \tag{10}$$

$$\mathcal{F}_2(x \cdot y) = \frac{1}{\sqrt{2\pi}} \cdot \mathcal{F}_2 x * \mathcal{F}_2 y \tag{11}$$

$$\begin{array}{ll}
\text{derivative} & \mathcal{F}_2(x') = -i \cdot \text{id} \cdot \mathcal{F}_2 x
\end{array} \tag{12}$$

3. Oscillations with period 2π and no roots of π

$$\text{FOURIER forward} \quad \mathcal{F}_3 x(\tau) = \int_{\mathbb{R}} \exp(i \cdot \tau \cdot t) \cdot x(t) \, dt$$

$$\text{FOURIER backward} \quad \mathcal{F}_3^{-1} x(\tau) = \frac{1}{2\pi} \cdot \int_{\mathbb{R}} \exp(-i \cdot \tau \cdot t) \cdot x(t) \, dt$$

We do not further consider Definition 3, because forward and backward FOURIER transform have different factors, thus laws for forward and backward transform differ in factors. For definitions 1 and 2 many laws are equal up to the choice of the transform direction.

At the first glance the definitions 1 and 2 of the FOURIER transform seem to be equally convenient or equally inconvenient. Thus many textbooks just introduce a definition and do not explain, why they prefer the one to the other possible ones. However, we think that factors in laws are bad, especially worse

than factors that can be hidden in a function definition. E.g. compare the eigenfunction property of the GAUSSIAN function in (7) and (9): We can define the GAUSSIAN bell curve in both ways and call it just g . The factor π does no longer get in the way when transforming equations containing \mathcal{F} and g . The same applies to the laws on derivatives in (8) and (12), where we can consider $2\pi i \cdot \text{id}$ as one function. In contrast to that, the convolution theorems in (10) and (11) have constant factors, even different ones. This makes manual equation manipulation cumbersome and error-prone. Even more it makes computations exclusively with rational numbers impossible, due to the irrational factor $\sqrt{2\pi}$. We could suppress the factors in (10) and (11) by defining convolution or multiplication containing a constant factor, but we think this is not natural.

For these reasons we will stick to definition 1 and omit the index of \mathcal{F} in the rest of this paper. Below we list the laws that follow from these definitions and that we want as axioms for our algebraic structure. Be aware, that we have omitted convergence conditions for the operations that involve integration. The laws marked with $\#$ can be derived from the remaining laws and are just given for the purpose of completeness.

$$\begin{array}{ll}
x + y = y + x & x + (y + z) = (x + y) + z \\
x \cdot y = y \cdot x & x \cdot (y \cdot z) = (x \cdot y) \cdot z \\
x * y = y * x \# & x \cdot (y + z) = x \cdot y + x \cdot z \\
& x * (y * z) = (x * y) * z \# \\
x \rightarrow 0 = x & x * (y + z) = x * y + x * z \# \\
x \downarrow 1 = x & (x \rightarrow a) \rightarrow b = x \rightarrow (a + b) \\
& (x \downarrow a) \downarrow b = x \downarrow (a \cdot b) \\
(x + y) \rightarrow a = (x \rightarrow a) + (y \rightarrow a) & (x \downarrow b) \rightarrow a = (x \rightarrow (a \cdot b)) \downarrow b \\
(x \cdot y) \rightarrow a = (x \rightarrow a) \cdot (y \rightarrow a) & (x + y) \downarrow a = (x \downarrow a) + (y \downarrow a) \\
(x * y) \rightarrow a = x * (y \rightarrow a) & (x \cdot y) \downarrow a = (x \downarrow a) \cdot (y \downarrow a) \\
(x \cdot y)' = x' \cdot y + x \cdot y' & (x * y) \downarrow a = |a| \cdot (x \downarrow a) * (y \downarrow a) \\
& (x * y)' = x * y' \\
\\
\mathcal{F}(x + y) = \mathcal{F}x + \mathcal{F}y & \mathcal{F}(k \cdot x) = k \cdot \mathcal{F}x \\
\mathcal{F}(x * y) = \mathcal{F}x \cdot \mathcal{F}y & \mathcal{F}(x \cdot y) = \mathcal{F}x * \mathcal{F}y \# \\
\langle x, y \rangle = \langle \mathcal{F}x, \mathcal{F}y \rangle & \|x\|_2 = \|\mathcal{F}x\|_2 \# \\
\mathcal{F}(x \downarrow a) = |a| \cdot \mathcal{F}x \downarrow \frac{1}{a} & \mathcal{F}(x \rightarrow a) = (\text{cis } \downarrow a) \cdot \mathcal{F}x \\
\mathcal{F}(\mathcal{F}x) = x \downarrow -1 & \\
\mathcal{F}(x^*) = \overline{\mathcal{F}x} & \mathcal{F}(x') = -2\pi i \cdot \text{id} \cdot \mathcal{F}x
\end{array}$$

3 Development of Gaussian models

Now that we have stated some axioms, we want to construct an ideal world, that is, a class of functions (or signals), where they hold. This class of function shall allow for simple constraints of the laws, for exact and efficient computation (more precisely: operations in a field), and shall allow to represent many mathematically important objects exactly and real world signals approximately.

Since GAUSSIAN functions are eigenfunctions of the FOURIER transform, they are perfect for representing signals in both time and frequency domain. They can be extended in a relatively simple way, such that all of the operations can be performed, that we listed initially.

We like to stress that GAUSSIAN functions are not only interesting, because they allow to perform the operations we want, but GAUSSIAN functions are central to signal processing and stochastics.

- GAUSSIAN functions are used as filter window for smoothing.
- GAUSSIAN functions “minimise uncertainty”, that is, they are the functions where HEISENBERG’s uncertainty relation becomes an equation. [7]
- With complex modulation GAUSSIAN functions are called GABOR atoms or *time-frequency atoms* in the GABOR transform. The GABOR transform is a windowed FOURIER transform, that shows how frequency components evolve over time.
- Complex modulated GAUSSIAN functions with a correction offset are called MORLET wavelets and are used in the Continuous Wavelet Transform. This transform is also intended for showing the evolution of frequency components over time, but it has higher time resolution and less frequency resolution for high frequencies.
- Best basis pursuits and matching pursuits are techniques for decomposing a signal into a finite number of irregularly located time-frequency atoms. [6]

Since GABOR transform, MORLET wavelet transform as well as best basis and matching pursuits aim at decomposition of a signal into time-frequency atoms, we have several tools for approximating real world signals using those atoms as building blocks.

In stochastics the density of the Normal distribution is a GAUSSIAN function. The Central Limit Theorem states, that adding more and more random variables (and divide by the square root of the number of added variables) in most practical cases approaches a normally distributed random variable. Translated to signal processing this means, that smoothing a signal again and again with practically any filter window, approximates a GAUSSIAN filter.

The derivation of the representations below is not particularly difficult, but we focus on finding representations that simplify most operations in terms of use of transcendental constants and irrational algebraic functions. We have implemented these function classes in Haskell using the NumericPrelude type class hierarchy. You find our implementation at

<http://code.haskell.org/numeric-prelude/src/MathObj/Gaussian/>.

3.1 Simple Gaussians

Simple GAUSSIAN functions shall be the first class of functions that we want to consider. In order to avoid a transcendental factor containing π in any of our function parameters when applying FOURIER transform \mathcal{F} , we cannot choose $f(t) = \exp(-t^2)$ but we have to choose its eigenfunction $f(t) = \exp(-\pi \cdot t^2)$. We also want to support shrinking of a function, what requires adding a shrinking parameter c , that we like to write in *curried form*: $f(c)(t) = \exp(-\pi \cdot (c \cdot t)^2)$. However this would yield a square root in the convolution of two such functions. It can be prevented by choosing the form $f(c)(t) = \exp(-\pi \cdot c \cdot t^2)$ with $c \in \mathbb{Q}$. The FOURIER transform of a shrunk function leads to another factor, the amplitude y of the function. We end up with the function form:

$$f(y, c)(t) = \sqrt{y} \cdot \exp(-\pi \cdot c \cdot t^2)$$

$$c \in \mathbb{Q} \quad y \in [0, \infty) \cap \mathbb{Q}.$$

This simple function class already allows for several operations, where convolution and FOURIER transform have constraints that assert convergence:

scaling	$k \cdot f(y, c) = f(y \cdot k^2, c)$
shrinking	$f(y, c) \downarrow k = f(y, c \cdot k^2)$
conjugate	$\overline{f(y, c)} = f(y, c)$
multiplication	$f(y_0, c_0) \cdot f(y_1, c_1) = f(y_0 \cdot y_1, c_0 + c_1)$
power with $r \geq 0$	$f(y, c)^r = f(y^r, r \cdot c)$
convolution $c_0 + c_1 > 0$	$f(y_0, c_0) * f(y_1, c_1) = f\left(\frac{y_0 \cdot y_1}{c_0 + c_1}, \frac{c_0 \cdot c_1}{c_0 + c_1}\right)$
FOURIER transform $c > 0$	$\mathcal{F}^{-1}(f(y, c)) = f\left(\frac{y}{c}, \frac{1}{c}\right).$

Typical functionals like function norms do not easily satisfy our goal of using the most simple algebraic structures. They need roots of parameters or constant transcendental factors, and thus require special treatment:

\mathcal{L}_1 -norm	$\ f(y, c)\ _1 = \sqrt{\frac{y}{c}}$
\mathcal{L}_2 -norm	$\ f(y, c)\ _2 = \sqrt{\frac{y}{\sqrt{2} \cdot c}}$
\mathcal{L}_∞ -norm	$\ f(y, c)\ _\infty = \sqrt{y}$
\mathcal{L}_p -norm	$\ f(y, c)\ _p = \sqrt[p]{\frac{y}{p \cdot c}}$
variance	$\frac{\ t \mapsto t^2 \cdot f(y, c)(t)\ _1}{\ f(y, c)\ _1} = \frac{1}{2\pi \cdot c}.$

3.2 Translated and modulated Gaussians

In the next step we want to translate and modulate the GAUSSIAN function. The most simple function class, that allows this, seems to be:

$$\begin{aligned} f(y, a, b, c)(t) &= \sqrt{y} \cdot \exp(-\pi \cdot (a + b \cdot t + c \cdot t^2)) \\ y &\in [0, \infty) \cap \mathbb{Q} \\ c &\in \mathbb{Q} \quad \{a, b\} \subset \mathbb{Q} + i\mathbb{Q}. \end{aligned} \quad (13)$$

With this representation we can perform the following operations:

$$\begin{aligned} \text{translation} \quad & f(y, a, b, c) \rightarrow k = f(y, a - b \cdot k + c \cdot k^2, b - 2 \cdot c \cdot k, c) \\ \text{modulation} \quad & f(y, a, b, c) \cdot (\text{cis1} \downarrow k) = f(y, a, b + 2 \cdot i \cdot k, c) \\ \text{scaling} \quad & k \cdot f(y, a, b, c) = f(y \cdot k^2, a, b, c) \\ \text{shrinking} \quad & f(y, a, b, c) \downarrow k = f(y, a, b \cdot k, c \cdot k^2) \\ \text{conjugate} \quad & \overline{f(y, a, b, c)} = f(y, \bar{a}, \bar{b}, c) \\ \text{power with } n \geq 0 \quad & f(y, a, b, c)^n = f(y^n, n \cdot a, n \cdot b, n \cdot c) \\ \text{multiplication} \quad & f(y_0, a_0, b_0, c_0) \cdot f(y_1, a_1, b_1, c_1) = f(y_0 \cdot y_1, a_0 + a_1, b_0 + b_1, c_0 + c_1) \\ \text{convolution } c_0 + c_1 > 0 \quad & f(y_0, a_0, b_0, c_0) * f(y_1, a_1, b_1, c_1) = \\ & f\left(\frac{y_0 \cdot y_1}{c_0 + c_1}, a_0 + a_1 - \frac{(b_0 - b_1)^2}{4 \cdot (c_0 + c_1)}, \frac{b_0 \cdot c_1 + b_1 \cdot c_0}{c_0 + c_1}, \frac{c_0 \cdot c_1}{c_0 + c_1}\right) \\ \text{FOURIER transform } c > 0 \quad & \mathcal{F}^{-1}(f(y, a, b, c)) = f\left(\frac{y}{c}, a - \frac{b^2}{4c}, -\frac{i \cdot b}{c}, \frac{1}{c}\right) \end{aligned}$$

The correctness of the equations for the FOURIER transform and the convolution are not so obvious. The FOURIER transform can be derived by translating the function to the origin and demodulate it, such that it becomes real. Then do FOURIER transform and translate and modulate it corresponding to the normalisations that we performed in time domain. The convolution can also be derived from such an normalisation. An alternative is to multiply in frequency domain.

We could also employ the definition

$$f(y, a, b, c)(t) = \sqrt{y} \cdot \exp(-(a + b \cdot \sqrt{\pi}t + c \cdot \pi t^2)) \quad (14)$$

and the formulas for most operations would remain the same, but translation, shrinking and modulation would have to be interpreted with respect to the unit $\sqrt{\pi}$.

With the considered representation we can also represent two other kinds of functions that are important to signal processing: A decaying exponential curve can be obtained with $c = 0 \wedge b > 0$. It is frequently encountered as envelope of percussive sounds. Unfortunately that curve is unrestricted in time, whereas in natural sounds the envelope usually starts suddenly somewhere in time.

The other important signal is a tone of linearly changing frequency, a *chirp*. In order to represent it, we have to generalise the real parameter c to a complex parameter. Setting $\Im(c) \neq 0, \Re(c) = 0$ yields a pure chirp, whereas $\Re(c) > 0$ yields a *chirplet*, that is a chirp, that is localised in time. The name is chosen analogously to *wavelets*. The FOURIER transforms and norm computations converge, if and only if, $\Re c > 0$ (i.e. only for chirplets), and the convolution exists if and only if $\Re(c_0 + c_1) > 0$. The BLUESTEIN transform allows us to write a FOURIER transform in terms of a convolution with a chirp.

$$\mathcal{F}^{-1}x = ((x \cdot f(1, 0, 0, -i)) * f(1, 0, 0, i)) \cdot f(1, 0, 0, -i)$$

The downside of this generalisation is, that we need to maintain a complex amplification factor y . This involves a complex square root and we must choose the right branch. For a single convolution or FOURIER transform choosing the branch with positive real part is just the right thing. The real part cannot vanish, since then the integrals do not converge at all. But when multiplying square roots with the convention of positive real parts then we must respect

$$\begin{aligned} \sqrt{c_0} \cdot \sqrt{c_1} &= (-1)^k \cdot \sqrt{c_0 \cdot c_1} \\ k &= \begin{cases} 1 : (\text{upper } c_0 \wedge \text{upper } c_1 \wedge \neg \text{upper}(c_0 \cdot c_1)) \vee \\ \quad (\neg \text{upper } c_0 \wedge \neg \text{upper } c_1 \wedge \text{upper}(c_0 \cdot c_1)) \\ 0 : \text{otherwise} \end{cases} \\ \text{upper } c &= \Im c > 0 \vee (\Im c = 0 \wedge \Re c < 0). \end{aligned}$$

That is we must maintain the sign of the real part separately. In the representation (13) we can implement a flipped sign by adding i to b . However maintaining this sign means comparisons and thus would not work for generalisations from \mathbb{Q} to other fields, e.g. finite fields.

3.3 Gaussians multiplied with polynomials

Another important operation in signal processing is derivation, be it in differential equations like oscillation equations, for the representation of an eigenbasis of the FOURIER transform, or as a *highpass filter*, that is, a frequency filter that emphasises high frequencies and suppresses low frequencies. Derivation requires, that we extend our representation to a product of a GAUSSIAN function and a polynomial function. However using the simple representation $f(y, a, b, c)(t) = \sqrt{y} \cdot \exp(-\pi \cdot (a + b \cdot t + c \cdot t^2))$ from (13), this would mean to maintain polynomial expressions of π as coefficients of the polynomial factor.

We like to avoid that and instead extend (14) to:

$$\begin{aligned}
f(y, a, b, c)(t) &= \sqrt{y} \cdot \exp(-(a + b\sqrt{\pi} \cdot t + c \cdot \pi \cdot t^2)) \\
\varphi((y, a, b, c), p)(t) &= f(y, a, b, c)(t) \cdot \widehat{p}(t) \\
y &\in [0, \infty) \cap \mathbb{Q} \\
c &\in \mathbb{Q} \quad \{a, b\} \subset \mathbb{Q} + i\mathbb{Q} \\
\widehat{p}(t) &\in (\mathbb{Q} + i\mathbb{Q})[\sqrt{\pi} \cdot t] \\
\widehat{p}(t) &= \sum_{j=0}^n p_j \cdot (\sqrt{\pi} \cdot t)^j
\end{aligned}$$

As mentioned above translation and modulation have to be interpreted with respect to a unit $\sqrt{\pi}$, and derivation must contain a factor of $\sqrt{\pi}$. However in order to get the usual units we can still replace \mathbb{Q} by $\mathbb{Q}(\sqrt{\pi})$.

For the following list of instantiations of signal processing transforms we like to subsume the parameters of f in a parameter tuple α .

$$\begin{aligned}
\text{translation} \quad & \varphi(\alpha, p) \rightarrow \sqrt{\pi}k = (f\alpha \rightarrow \sqrt{\pi}k) \cdot (\widehat{p} \rightarrow \sqrt{\pi}k) \\
\text{modulation} \quad & \varphi(\alpha, p) \cdot \left(\text{cis}1 \downarrow \frac{k}{\sqrt{\pi}} \right) = \left(f\alpha \cdot \left(\text{cis}1 \downarrow \frac{k}{\sqrt{\pi}} \right) \right) \cdot \widehat{p} \\
\text{scaling} \quad & k \cdot \varphi(\alpha, p) = f\alpha \cdot (k \cdot \widehat{p}) \\
\text{shrinking} \quad & \varphi(\alpha, p) \downarrow k = (f\alpha \downarrow k) \cdot (\widehat{p} \downarrow k) \\
\text{conjugate} \quad & \overline{\varphi(\alpha, p)} = \overline{f\alpha} \cdot \widehat{\overline{p}} \\
\text{multiplication} \quad & \varphi(\alpha_0, p_0) \cdot \varphi(\alpha_1, p_1) = (f(\alpha_0) \cdot f(\alpha_1)) \cdot \widehat{p_0 \cdot p_1} \\
\text{power with } n \in \mathbb{N} \quad & \varphi(\alpha, p)^n = (f\alpha)^n \cdot \widehat{p^n} \\
\text{convolution} \quad & \varphi(\alpha_0, p_0) * \varphi(\alpha_1, p_1) = \mathcal{F}(\mathcal{F}^{-1}(\varphi(\alpha_0, p_0)) \cdot \mathcal{F}^{-1}(\varphi(\alpha_1, p_1)))
\end{aligned}$$

FOURIER transform

$$\begin{aligned}
\mathcal{F}^{-1}(\varphi(\alpha, s : p)) &= s \cdot \mathcal{F}^{-1}(f\alpha) + \frac{i}{2 \cdot \sqrt{\pi}} \cdot (\mathcal{F}^{-1}(\varphi(\alpha, p)))' \\
\text{where } \overline{s} : \widehat{p}(t) &= s + \sqrt{\pi} \cdot t \cdot \widehat{p}(t)
\end{aligned}$$

Differentiation

$$\frac{1}{\sqrt{\pi}} \cdot (\varphi((y, a, b, c), p))' = f(y, a, b, c) \cdot \left(\frac{1}{\sqrt{\pi}} \cdot \widehat{p}' - (t \mapsto b + c \cdot \sqrt{\pi} \cdot t) \cdot \widehat{p} \right) \quad (15)$$

Integration

$$\begin{aligned}
& \sqrt{\pi} \cdot \int_{-\infty}^T \varphi((y, a, b, c), p)(t) \, dt \\
&= s \cdot \sqrt{\pi} \cdot \int_{-\infty}^T f(y, a, b, c)(t) \, dt + f(y, a, b, c)(T) \cdot \widehat{q}(T) \\
&= s \cdot \exp\left(-a + \frac{b^2}{4c}\right) \cdot \frac{1 + \text{erf}\left(\frac{b}{2\sqrt{c}} + \sqrt{c\pi} \cdot T\right)}{2\sqrt{c}} \\
&\quad + f(y, a, b, c)(T) \cdot \widehat{q}(T) \\
&\text{where } \widehat{q}(t) = \frac{\frac{1}{\sqrt{\pi}} \cdot \widehat{q}'(t) - \widehat{p}(t) - s}{b + c \cdot \sqrt{\pi} \cdot t} \quad (16)
\end{aligned}$$

Eigenfunction of FOURIER transform

$$e_n = f(1, 0, 0, 2)^{(n)} \cdot f(1, 0, 0, -1) \quad (17)$$

The equation (16) is the inverse of (15). This implies that in (16) the polynomial q depends recursively on itself. However because the degree of p is one more than that of q , the leading term of q only depends on the leading term of p . Thus we can successively determine the terms of q starting at the highest one. The equation can be translated almost literally to a polynomial division with remainder s in our Haskell implementation and be solved using *lazy evaluation*.

In (17) we have used the definition of HERMITE polynomials and the known fact, that the GAUSSIAN function multiplied with a HERMITE polynomial is an eigenfunction of the FOURIER transform.

3.4 Mixed Gaussians

In order to support sums of signals, we must maintain a set A of parameters for the GAUSSIANS and a map P from the GAUSSIAN parameters α to the associated polynomial factor.

$$\sum_{\alpha \in A} f\alpha \cdot \widehat{P\alpha}$$

Eventually, this representation is general enough in order to be target of a windowed FOURIER transform or a best basis or matching pursuit. That is we can approximate real world signals in a natural way and perform exact signal processing operations on them.

4 Related work

With our paper we wanted to draw a connection between Computer Algebra on the one side and Signal Processing and Stochastics on the other side. With “Computer Algebra” we mean exact computations involving complex mathematical objects like polynomials, polynomial ideals, groups, that is at a higher level than computing with individual numbers but at a lower level than computing with general mathematical expressions as in symbolic manipulations. Although signal processing is certainly not the most prominent application of computer algebra, there are many problems that were solved using computer algebra methods. [4] The most famous application is probably the Discrete FOURIER Transform, that can be performed in log-linear time with respect to the length of the input data using techniques from number theory, finite fields, polynomial rings and automated code generation. [1,3] Closely related is the fast convolution of discrete signals, that uses the Fast FOURIER Transform and by the chirp transform it is also possible to express a fast FOURIER Transform in terms of fast convolution algorithms. Another computer algebra application in signal processing is the design of frequency filters, where we have to construct rational functions given conditions for the location of its zeros and poles. [5]

In [2] the authors develop signal processing the algebraic way, as we do in our paper. That is, opposed to the sample value focus of most signal processing literature they treat signals as objects, define operations on them and propose and prove laws. The book is concerned with two-dimensional signals, but the difference to one-dimensional signals is not essential in this approach. Unlike our consideration of real functions and especially modified GAUSSIAN functions the authors stick to discrete signals.

Compared to established computer algebra systems and their symbolic integration machineries, our framework provides no new class of closedly integrable functions. All of Maple 10, Mathematica 5.2, Maxima 5.20.1 can integrate the integrals occurring in convolution, FOURIER transform, norm and scalar product of our products of Gaussian function and polynomial in closed form. MuPAD 4.0.6 cannot integrate the more complicated convolutions and Axiom 20091101 cannot cope with the integrals at all, since it does not yet support assumptions. We need assertion for the coefficient c of the quadratic term in the exponent of the GAUSSIAN function. It must have positive real part, or must be at least a positive real number. That is, with an assertion we must exclude signals with constant amplitude or even unbounded amplitude.

5 Outlook

5.1 Dirac impulses

The DIRAC impulse δ is a virtual function that is infinitely high at the origin and zero elsewhere, enclosing an infinitely high and infinitely narrow rectangle of area 1. If this function would exist, it would be the identity element of convolution. Its FOURIER transform is the function that is constant 1, because this is the identity element of pointwise function multiplication. In stochastics the DIRAC impulse is needed for representing mixed discrete/continuous probability distributions. Existence of derivatives of the DIRAC impulse would eliminate the need for a distinct differentiation operation, because $x' = x * \delta'$. We could also more easily hide the factor $\sqrt{\pi}$ in the differentiation operation and we could represent frequency spectra of polynomial functions.

Several approaches like SCHWARTZ distributions and non-standard analysis were developed, in order to get a strictly founded notion of a DIRAC impulse. However, none of them is completely satisfying: SCHWARTZ distributions have no longer a notion of function application and they cannot be multiplied pointwise. Non-standard analysis allows to define infinitely high and infinitely narrow functions, that actually let real functions unaltered at the “coarse” scale of real numbers after convolution. However when considering a non-standard function on all scales, convolution with the non-standard DIRAC impulse well changes the convolution partner. It is not known to us, whether an approach can exist at all, that fulfils all expectations.

All the more it is interesting whether we can have an object, that exactly behaves like a DIRAC impulse in our theory. Since our theory is abstracted from,

but not bound to real functions, we could check this way, whether a DIRAC impulse makes sense at all. Formally in our approach a DIRAC impulse could be represented by $f(1, 0, 0, +\infty)$. The term $+\infty$ could be made precise by using projective geometry, i.e. by allowing an object like $\frac{1}{0}$. But then it is open, whether we should use

1. $\exp\left(-\pi \cdot \frac{a+b \cdot t+c \cdot t^2}{d}\right),$
2. $\exp\left(-\pi \cdot \left(a+b \cdot t+\frac{c}{d} \cdot t^2\right)\right),$
3. $\exp\left(-\pi \cdot \left(a+b \cdot \frac{t}{d}+c \cdot \frac{t^2}{d^2}\right)\right)$ or
4. $\exp\left(-\pi \cdot \frac{a_0}{a_1}+\frac{b_0}{b_1} \cdot t+\frac{c_0}{c_1} \cdot t^2\right)$

with a projective interpretation of the fractions, and how to cope with the amplitude parameter.

5.2 Discrete signal processing

We would like to have the same set of operations and laws for discrete signals that we already have for real signals. In order to have dual time and frequency domains, we need to content ourselves with periodic discrete signals. For instance for discrete periodic signals x and y of period length n convolution and Discrete FOURIER Transform (DFT) are usually defined as:

$$\begin{aligned}(x * y)_k &= \sum_{j \in \mathbb{Z}_n} x_j \cdot y_{k-j} \\ (\text{DFT } x)_k &= \frac{1}{\sqrt{n}} \cdot \sum_{j \in \mathbb{Z}_n} \exp\left(\frac{2\pi i}{n} \cdot j \cdot k\right) \cdot x_j \\ (\text{DFT}^{-1} x)_k &= \frac{1}{\sqrt{n}} \cdot \sum_{j \in \mathbb{Z}_n} \exp\left(-\frac{2\pi i}{n} \cdot j \cdot k\right) \cdot x_j.\end{aligned}$$

The factor $\frac{1}{\sqrt{n}}$ is chosen, such that DFT becomes unitary. However with this definition it does not hold $\text{DFT}(x \cdot y) = \text{DFT } x * \text{DFT } y$, but instead $\sqrt{n} \cdot \text{DFT}(x \cdot y) = \text{DFT } x * \text{DFT } y$. One solution would be to add the factor $\frac{1}{\sqrt{n}}$ to the definition of the convolution. This is at least very uncommon. An alternative is to treat discrete signals as piecewise constant functions. The sums are turned to integrals and thus need a step width. To this end we equip every signal with a *sampling rate* and denote it with *rate*. It holds $\text{rate}(\text{DFT } x) = \frac{n}{\text{rate } x}$. We obtain the definitions

$$\begin{aligned}(x * y)_k &= \frac{1}{\text{rate } x} \cdot \sum_{j \in \mathbb{Z}_n} x_j \cdot y_{k-j} \quad \text{for } \{\text{rate } x, \text{rate } y\} = \{\text{rate}(x * y)\} \\ (\text{DFT } x)_k &= \frac{1}{\text{rate } x} \cdot \sum_{j \in \mathbb{Z}_n} \exp\left(\frac{2\pi i}{n} \cdot j \cdot k\right) \cdot x_j \\ (\text{DFT}^{-1} x)_k &= \frac{1}{\text{rate } x} \cdot \sum_{j \in \mathbb{Z}_n} \exp\left(-\frac{2\pi i}{n} \cdot j \cdot k\right) \cdot x_j.\end{aligned}$$

Following the reasoning of real signals, we need eigenfunctions of the FOURIER transform. Generic simply representable eigenbases of the Discrete FOURIER transform are currently not known, but according to the POISSON summation formula, the discretised and periodically summed eigenfunctions of the Continuous FOURIER transform, are eigenvectors of the discrete transform. However discretising an eigenbasis of the continuous FOURIER transform may not yield a discrete eigenbasis.

In discrete signal processing the identity element of convolution is simple to get: It is the signal that is 1 at index 0 and zero elsewhere. In contrast to that, the operation of signal dilation may lead to undefined and multiple times defined elements in the resulting vector. The natural solution is to set undefined elements to zero and sum up all candidates for multiply defined output elements. This can be written generally in the following way, where n is the signal period length and the empty sum is zero:

$$(x \uparrow k)_j = \sum_{l \in \mathbb{Z}_n: l \cdot k = j} x_l.$$

This definition matches shrinking the vector in the frequency domain. Nonetheless, we have to drop invertibility of dilation from the list of laws, that hold for real signals.

Another problem is the definition of differentiation. We could replace it by centred discrete differences. Via the FOURIER transform this would also yield a notion of periodic polynomials, namely polynomials in $\sin \frac{2\pi \cdot t}{n}$ instead of t . But this interpretation of differentiation is different from differentiation of a continuous function with subsequent discretisation and periodic summation. Thus it cannot be used for eigenvector computation in the same way we used it for continuous signals.

Summarised, we cannot simply perform the operations on our parameter tuples, that we developed for continuous signals, and use them for discrete signals by just interpreting them in a discrete way.

References

1. Clausen, M., Baum, U.: Fast Fourier Transforms. BI Wissenschaftsverlag (1993)
2. D'Alotto, L.A., Giardina, C.R., Luo, H.: A unified signal algebra approach to two-dimensional parallel digital signal processing. Monographs and textbooks in pure and applied mathematics, Marcel Dekker (1998)
3. Frigo, M., Johnson, S.G.: FFTW: The Fastest FOURIER Transform in the West, version 3. <http://www.fftw.org/>
4. Grabmeier, J., Kaltofen, E., Weispfenning, V. (eds.): Computer Algebra Handbook. Springer (2003)
5. Hamming, R.W.: Digital Filters. Signal Processing Series, Prentice Hall (January 1989)
6. Mallat, S.G., Zhang, Z.: Matching pursuits with time-frequency dictionaries. IEEE Transactions on Signal Processing 41(12), 3397–3415 (December 1993), <http://ieeexplore.ieee.org/search/wrapper.jsp?arnumber=258082>
7. Triebel, H.: Höhere Analysis. Harri Deutsch (1980)